



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant: Kiesel, et al

Serial No.: 09/757,146

RECEIVED

FEB 04 2002

Group 2100

Title: Universal Motion Controller

Examiner: Unknown

Docket No.: 2000 P 03001 US

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service on the date indicated below with sufficient postage as First class mail in an envelope addressed to: Assistant Commissioner for Patents, BOX DAC, Washington, DC 20231

Date: 1/22/02
Nancy A. Demko

SUBMISSION OF PRIORITY DOCUMENT

Assistant Commissioner for Patents
Washington, DC 20231

Sir:

Applicant hereby submits the enclosed priority documents No. 100 00 626.4 in the above referenced patent application, filed on November 23, 2000. Please make these documents of record in this case.

Respectfully submitted,
I. Marc Asperas, Esq.
Reg. No. 37,274

Date: 1/22/02

Siemens Corporation
Intellectual Property Department
186 Wood Avenue South
Iselin, New Jersey 08830
(732) 321-3009

RECEIVED

APR 29 2002

OFFICE OF PETITIONS

BUNDESREPUBLIK DEUTSCHLAND



RECEIVED
FEB 04 2002
Group 2100

Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 100 00 626.4
Anmeldetag: 10. Januar 2000
Anmelder/Inhaber: Siemens Aktiengesellschaft,
München/DE
Bezeichnung: Universelle Bewegungssteuerung
IPC: G 05 B 19/02

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 23. November 2000
Deutsches Patent- und Markenamt
Der Präsident
Im Auftrag

Waasmaier

RECEIVED
APR 29 2002
OFFICE OF PETITIONS

Beschreibung

Universelle Bewegungssteuerung

5 Die Erfindung bezieht sich auf eine universelle Bewegungs-
steuerung mit Engineering- und Run-Time-System, welche funk-
tionell die klassischen Aufgaben einer speicherprogrammierba-
ren Steuerung und einer numerischen Steuerung in sich verei-
nigt.

10

Es ist heutzutage üblich, sowohl für die speicherprogrammier-
bare Steuerung als auch für die Bewegungssteuerung jeweils
unterschiedliche hierarchische Ablaufebenen zu modellieren,
denen Software-Tasks zur Steuerung des jeweiligen technischen
15 Prozesses zugeordnet werden. Diese Tasks können Systemaufga-
ben erfüllen, sie können aber auch anwenderprogrammiert sein.

20

Es ist bekannt, daß bei einer speicherprogrammierbaren Steue-
rung "SPS", also auch bei einer Bewegungssteuerung "NC", An-
wenderprogramme bzw. vom Anwender erstellte Tasks in den
20 Speicher der jeweiligen Steuerung zugeladen und zur Ausfüh-
rung gebracht werden können.

25

Aus der DE 197 40 550 A1 ist es bekannt, daß Prozeßsteue-
rungsfunktionalitäten der speicherprogrammierbaren Steuerun-
gen "SPS" und Bewegungsfunktionalitäten von NC-Steuerung in
einem einheitlichen konfigurierbaren Steuerungssystem integ-
riert werden können.

30

Diese SPS/NC-Integration geschieht in Form des Zusammenschal-
tens von SPS- und NC-Steuerungsbaugruppen. Bei einer solchen
Ausführung der Integration wird aber keine optimale und effi-
ziente Taskstruktur für die Gesamtheit der Steuerungsaufgaben
erreicht. Außerdem kann bezüglich der Prozeßsteuerung, also
35 auch bezüglich der Bewegungssteuerung, erweiternde Funktiona-
lität nur in Form von Anwenderprogrammen nachgeladen und zur
Ausführung gebracht werden.

Es ist heutzutage üblich, Steuerungen mit Parametrierinformationen zu versorgen.

Parametrierinformationen umfassen in diesem Zusammenhang

- 5 - die Beschreibung von Systemvariablen mit Datentyp, Attributen und Beschreibungstexten,
- die Beschreibung von Alarmen, mit ihrem strukturellen Aufbau, Attributen und Alarmtexten, sowie
- 10 - die Beschreibung von Sprachbefehlen (Bewegungs- und Technologiebefehle) mit Syntax und dazugehörigen Parametern.

Üblicherweise werden aber diese Parametrierinformationen, die an unterschiedlichen Stellen innerhalb der Steuerung benötigt werden, jeweils separat an diesen Stellen in der Steuerung
15 implementiert. Die Konsistenz dieser verteilt implementierten Parametrierinformationen ist nur sehr aufwendig sicherstellbar.

Der Erfindung liegt daher die Aufgabe zugrunde, für jeweils
20 unterschiedliche Steuerungsaufgaben und unterschiedliche Randbedingungen bzw. Anforderungen des zugrundeliegenden technischen Prozesses in einfacher Weise optimale Ausprägungen der kombinierten SPS/NC-Steuerungen sowohl hinsichtlich ihrer Steuerungsstruktur als auch hinsichtlich ihrer Funktio-
25 nalität zu erstellen und dabei sicherzustellen, daß die in der Steuerung implementierten Parametrierinformationen immer konsistent zueinander sind.

Gemäß der Erfindung wird diese Aufgabe dadurch gelöst, daß
30 ein einheitliches Ablaufebenenmodell dergestalt gebildet ist, daß es mehrere Ablaufebenen unterschiedlichen Typs mit unterschiedlicher Priorität aufweist, wobei von höchster bis niedrigster Priorität verschiedene Anwender- und Systemebenen vorgesehen sind und daß jeweils Technologiepakete anwender-
35 seitig in das Engineering- und/oder Run-Time-System ladbar sind, daß eine Datenquelle für Beschreibungsinformationen für Systemvariablen sowie gegebenenfalls Alarme und/oder Sprach-

befehle über einen Umsetzer dem Engineering-System Sprachbefehle und/oder Systemvariablen zur Verfügung stellt, daß aus dem Run-Time-System die Systemvariablen mit aktuellen Daten des technischen Prozesses versorgbar sind und daß über eine
5 Bedienoberfläche des Engineering-Systems weitere Eingaben anwenderseitig machbar sind.

Ein wesentlicher Vorteil der Erfindung liegt darin, daß die Parametrierinformationen der Steuerung, die sowohl im Engineering-System, im Run-Time-System, aber auch für die Dokumentation und eine etwaige Testautomatisierung benötigt werden, immer konsistent sind. Der Umsetzer, der an zentraler Stelle die Parametrierinformationen für die Dokumentation, das Engineering-System und das Run-Time-System aufbereitet
10 und verteilt, kann ohne großen Aufwand Semantikchecks durchführen. Auch können OEM (Original Equipment Manufacturer)-Kunden in dieser Datenquelle für Beschreibungsinformationen, d.h. an definierter Stelle, aufwandsarm weitere Parametrierinformationen für die Steuerung erstellen und in die Dokumentation einbringen.
15 20

Ein weiterer Vorteil der Erfindung liegt darin, daß innerhalb der Ablaufebenen die Tasks der Steuerung so angeordnet werden können, daß der Kommunikationsaufwand innerhalb der Steuerung
25 reduziert wird.

Ein weiterer Vorteil liegt darin, daß anwenderseitig durch die Zuladung von Technologiepaketen in das Engineering- und/oder Run-Time-System der Steuerung eine anwendungsspezifische Skalierung des Run-Time-Systems der Steuerung bezüglich ihrer Funktionalität erreicht werden kann.
30

Eine erste vorteilhafte Ausgestaltung der Erfindung liegt darin, daß vom Umsetzer aus dem Bestand der Datenquelle relevante Dokumentationsinformationen an ein Ausgabemedium weiterleitbar sind. Dadurch wird sichergestellt, daß alle Dokumentationsinformationen aus einer gemeinsamen Datenquelle
35

stammen und somit immer untereinander konsistent sind, unabhängig, auf welches Ausgabemedium (z.B. Drucker oder Online-Hilfe) die Dokumentationsinformation ausgegeben wird.

5 Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, daß als Ablaufebenen vorgesehen sind:

a) eine Lagereglerebene, bestehend aus zugehöriger getakteter Systemebene und Anwenderebene,

10

b) eine Interpolatorebene, bestehend aus zugehöriger getakter Systemebene und Anwenderebene,

c) eine Event-Systemebene für reaktionspflichtige Ereignisse,

15

d) eine Anwenderebene für asynchrone Fehler,

e) eine weitere, vom Anwender anforderungsspezifisch frei projektierbare Anwenderebene für Alarm- und/oder Event- und/oder Regelungs- und/oder sonstige zyklische Tasks,

20

f) eine aus der Abfolge von Bewegungssequenzen, freien Zyklen und niederprioren sonstigen Systemtasks, gebildete Ebenengruppe für Hintergrund-Bearbeitung,

25

wobei die Ablaufebenen a bis e eine Ebenengruppe für Echtzeit-Bearbeitung bilden.

Ein wesentlicher Vorteil dieser Schichtung liegt darin, daß die Kommunikation zwischen den Tasks der Prozeßsteuerung und denen der Bewegungssteuerung minimiert wird. Dadurch kann die Programmierung der Steuerungsaufgaben für die Prozeßsteuerung und für die Bewegungssteuerung in einer einheitlichen Programmiersprache mit einer einheitlichen Erstelloberfläche erfolgen.

35

Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, daß die Technologiepakete beinhalten:

a) Code-Teile, die die Regelungsspezifika für das Run-Time-System repräsentieren und

5 b) einen Konfigurierteil der die Zuordnung dieser Code-Teile zu den jeweiligen Systemebenen, sowie deren Bearbeitungsfolge aufweist, wobei

10 c) bedarfsweise diese Informationen des Konfigurierteils auch an das Engineering-System weiterleitbar sind.

Somit ist es möglich, daß der Anwender durch dynamisches Zuladen solcher Technologiepakete die Möglichkeit einer technologischen Skalierung des Run-Time-Systems der Steuerung besitzt. Damit kann der Anwender ausgehend von einem Basissystem der Steuerung den Befehlsvorrat dieses Basissystems bzw. Betriebssystems dynamisch zugeschnitten auf die jeweiligen Erfordernisse des zugrundeliegenden technologischen Prozesses oder der Steuerungsaufgabe erweitern. Der Anwender hat somit
15 die Möglichkeit, die vorhandene Grundfunktionalität einer Steuerung gezielt um solche Funktionalitäten zu erweitern, die er wirklich für seine Anwendungen benötigt. Das Basissystem bildet hierbei den Auslieferungsumfang des Run-Time-Systems einer Steuerung, nämlich ein Echtzeitbetriebssystem,
20 ein Ablaufsystem (mit System- und Anwenderebenen), Technologieobjekttypen, Sprachbefehle, den SPS-Befehlsvorrat sowie Kommunikations- (z.B. LAN, E/A) und technologische Schnittstellen (z.B. Antriebe, Geber) zum technischen Prozeß. Im Basissystem befindet sich somit die notwendige Grundfunktionalität einer Steuerung. Das Basissystem ist dabei auf unterschiedlichsten HW-Plattformen (z.B. PC, Antrieb, ...) ablauffähig.
25
30

Dadurch, daß die Informationen des Konfigurierteils eines
35 Technologiepaketes über die Datenquelle und den Umsetzer ins Run-Time-System und ins Engineering-System gebracht werden, lassen sich Parametrier- und Konfigurierinformationen in ein-

heitlicher Weise in die Steuerung einbringen und ein Anwender kann an zentraler Stelle Änderungen in den Parametrier- bzw. Konfigurierdaten vornehmen. Die sogenannten Parametrierinformationen entsprechen Datenbeschreibungen für übliche und generelle Steuerungsaspekte, nämlich Systemvariablen, Alarmen und Sprachbefehlen. Die Konfigurierinformationen beziehen sich hingegen auf Technologiepakete und damit auf die Möglichkeit einer technologischen Skalierung einer Steuerung.

10 Dadurch, daß jedes Technologiepaket eine angepaßte Anzahl von Technologieobjekttypen für das Run-Time-System beinhaltet, ist es möglich, auch komplexe und anspruchsvolle Steuerungsfunktionalitäten in einer übersichtlichen und verständlichen Form dem Run-Time-System zuzuladen.

15 Eine weitere vorteilhafte Ausgestaltung der Erfindung liegt darin, daß des weiteren Bedienoberflächeninformationen, insbesondere Bedienparameter, und/oder Sprachmechanismen und/oder Deklarationsteile den Code-Teilen zuweisbar sind.

20 Dadurch ergeben sich folgende Vorteile:

25 Um einen Technologieobjekttyp nicht nur als nicht mehr änderbare Konstante verwenden zu können, muß der Technologieobjekttyp dem Erstellsystem die Möglichkeiten der Parametrierung für seine instanziierten Technologieobjekte und insbesondere die vorhandenen Bedienparameter bekanntmachen. Damit hat ein Anwender die Möglichkeit, ein Technologieobjekt in der Oberfläche des Erstellsystems flexibel zu parametrieren.

30 Dadurch, daß auch Sprachmechanismen zuladbar sind, ist es möglich, daß dynamisch der Befehlsvorrat des Run-Time-Systems erweitert werden kann. In einem Anwenderprogramm kann der Anwender einen solchen zugeladenen Befehl in der Form verwenden, als wäre er ein Befehl der Grundfunktionalität.

35 Wenn ein Anwenderprogramm mit einem solchen zugeladenen Befehl innerhalb einer Anwenderenebene des Ablaufebenenmodells

abgearbeitet wird, wird bei Aufruf dieses zugeladenen Befehls die dazugehörige Codesequenz des Betriebssystems auf einer der Systemebenen des Ablaufebenenmodells abgearbeitet. Dies geschieht ohne Zutun des Anwenders. Durch die Zuordnung von Deklarations- und Beschreibungsteilen zu den Code-Teilen des Technologiepaketes wird die Flexibilität für den Anwender weiterhin erhöht.

Ein Ausführungsbeispiel der Erfindung ist in der Zeichnung dargestellt und wird im folgenden erläutert.

Dabei zeigen:

- 15 FIG 1 die Steuerung eines technischen Prozesses mit getrennter speicherprogrammierbarer Steuerung und Bewegungssteuerung. Die Programmierung erfolgt über jeweils separate Programmiersysteme,
- 20 FIG 2 die wesentlichen Ablaufebenen einer klassischen speicherprogrammierbaren Steuerung,
- 25 FIG 3 die wesentlichen Ablaufebenen einer Bewegungssteuerung,
- 30 FIG 4 eine universelle Steuerung, d.h. eine kombinierte SPS/NC-Steuerung mit einem dazugehörigen Programmiersystem,
- 35 FIG 5 das Ablaufebenenmodell der universellen Steuerung,
- FIG 6 zeigt als OO(objektorientiert)-Strukturdiagramm ein Technologiepaket, bestehend aus Code-Anteil, Parameter, Firmware-Konfiguration, Technologieobjekttyp, Sprachkomponente und Deklarationsteil,
- FIG 7 zeigt als OO-Strukturdiagramm Technologieobjekttypen für das Technologiepaket Kunststoff und

FIG 8 zeigt, wie aus einer Datenquelle die Beschreibungs- bzw. Parametrierinformationen über einen Umsetzer dem Engineering-System, dem Run-Time-System und einem Ausgabemedium zur Verfügung gestellt werden.

In der Darstellung gemäß FIG 1 wird, in Form eines Strukturbildes gezeigt, daß zur Steuerung eines technischen Prozesses TP1 ein paralleler Betrieb einer speicherprogrammierbaren Steuerung SPS und einer Bewegungssteuerung NC stattfindet. Speicherprogrammierbare Steuerung SPS und Bewegungssteuerung NC enthalten jeweils ein Run-Time-System RTS1 bzw. RTS2. Die Kommunikation zwischen den beiden Steuerungen erfolgt über spezielle Hilfsmittel, exemplarisch dargestellt ist ein bidirektionaler Kommunikationskanal K. Die Programmierung der Steuerungen durch den Anwender erfolgt in der Regel in unterschiedlichen Programmiersprachen mit unterschiedlichen Erstelloberflächen. Das heißt, durch jeweils separate Programmier- oder Engineering-Systeme P1, ES1 und P2, ES2. Der wesentliche Nachteil dieser konventionellen Ausführung liegt zum einen in der aufwendigen Kommunikation zwischen den beiden Steuerungen, zum anderen in den separaten und unterschiedlichen Programmier- bzw. Engineering-Systemen P1, ES1 und P2, ES2. Über Ein- und Ausgänge EA1, EA2 der Steuerungen wird der eigentliche technische Prozeß TP1 gesteuert. Zwischen dem Programmiersystem P1 und der speicherprogrammierbaren Steuerung SPS bzw. zwischen dem Programmiersystem P2 und der numerischen Steuerung NC befinden sich Informationspfade I1 bzw. I2, auf denen die Programme in die jeweilige Steuerung geladen werden.

In der Darstellung gemäß FIG 2 sind die wesentlichen Ablaufebenen einer klassischen speicherprogrammierbaren Steuerung (SPS; FIG 1), angeordnet nach ihrer Priorität, gezeigt. Der Prioritätsanstieg ist dabei durch einen Pfeil symbolisiert. In der niederpriorsten Ebene werden, wie durch eine gestrichelte Linie angedeutet, zwei unterschiedliche Aufgaben, näm-

lich ein freier Zyklus, d.h. "Anwenderebene freier Zyklus" und eine Hintergrund-Systemebene, d.h. "Systemebene Hintergrund", im Round-Robin-Verfahren, also zeitscheibengesteuert, abgewickelt. Der Hintergrund-Systemebene sind z.B. Kommunikationsaufgaben zugeordnet. Bei einer folgenden getakteten Anwenderebene, bezeichnet als "Anwenderebene getaktet", ist der Aufruftakt der Tasks bzw. der Programme dieser Ebene parametrierbar. Es erfolgt eine Überwachung dahingehend, ob die Bearbeitung eines Anwenderprogrammes dieser getakteten Ebene rechtzeitig abgeschlossen ist, bevor das Startereignis erneut auftritt. Läuft die Taktzeit ab, ohne daß das Anwenderprogramm der zugeordneten Ebene fertig abgearbeitet ist, wird eine entsprechende Task einer prioritätsmäßig übernächsten "Anwenderebene für asynchrone Fehler" gestartet. In dieser "Anwenderebene für asynchrone Fehler" kann der Anwender die Behandlung von Fehlerzuständen ausprogrammieren.

Auf die "Anwenderebene getaktet" folgt eine "Anwenderebene Events". Die Reaktion auf externe oder interne Ereignisse (Events) erfolgt innerhalb der "Anwenderebene Events". Ein typisches Beispiel für ein solches Ereignis ist das Überschreiten eines Grenzwerts. In einer "Systemebene hochprior" liegen Aufgaben des Betriebssystems, welche die Arbeitsweise der speicherprogrammierbaren Steuerung sicherstellen.

Die Darstellung gemäß FIG 3 zeigt die wesentlichen Ablaufebenen einer Bewegungssteuerung (NC; FIG 1). Auch hierbei sind die einzelnen Ebenen nach ihrer Priorität hierarchisch, wie durch einen Pfeil symbolisiert, angeordnet. Eine "Systemebene Hintergrund" und eine "Anwenderebene sequenziell" haben eine gleiche Priorität, nämlich die niedrigste. Diese aufgabenmäßige Zusammengehörigkeit ist wie bei FIG 2 durch eine gestrichelte Linie symbolisiert. Die Tasks der "Anwenderebene sequenziell" werden zusammen mit den Tasks der "Systemebene Hintergrund" im Round-Robin-Verfahren abgearbeitet. Typische Tasks der "Systemebene Hintergrund" sind z.B. solche für Kommunikationsaufgaben. In der "Anwenderebene sequenziell" lau-

fen die vom Anwender programmierten Programmteile für die eigentliche Steuerungsaufgabe. Stößt die Steuerung in einem dieser Programmteile auf einen Bewegungs- oder Positionierbefehl, wird ein Suspend gesetzt, d.h. das Anwenderprogramm wird an dieser Stelle unterbrochen. Die Abarbeitung dieses Bewegungs- oder Positionierbefehls geschieht in einer höchstpriorigen "Systemebene getaktet". Ein jeder Lageregler, der in der "Systemebene getaktet" abläuft, führt diesen Bewegungs- bzw. Positionierbefehl aus. Nach Ausführung des Befehls wird in die "Anwenderebene sequenziell" zurückgesprungen und das durch Suspend unterbrochene Anwenderprogramm wird durch ein Resume an der gleichen Stelle fortgesetzt. Die "Systemebene getaktet" enthält neben den schon erwähnten Lagereglern auch den Interpolationsteil der Steuerung.

Auf die niederpriorste Ebene setzt eine "Anwenderebene getaktet" auf. Hier laufen zyklische Tasks ab, z.B. Reglerfunktionalitäten.

In einer folgenden "Anwenderebene Events" sind solche Tasks untergebracht, die auf externe oder interne Ereignisse reagieren. Solche Ereignisse können beispielsweise Alarme sein.

In der Darstellung gemäß FIG 4 wird ein technischer Prozeß TP2 durch eine kombinierte SPS/NC-Steuerung UMC gesteuert. Das Akronym UMC steht für UNIVERSAL-MOTION-CONTROL. Die Verbindung zwischen der Steuerung UMC und dem zugehörigen technischen Prozeß TP2 geschieht bidirektional über Ein-/Ausgänge EA3. Die Programmierung der kombinierten SPS/NC-Steuerung geschieht über ein gemeinsames Programmier- P3 oder Engineering-System ES3, wobei das Engineering-System ES3 ebenso wie bei FIG 1 eine komfortable Oberfläche für das Programmiersystem P3 zur Verfügung stellt. Die damit erstellten Programme werden über einen Informationspfad I3 in ein Run-Time-System RTS3 der universellen Bewegungssteuerung UMC übertragen. Die Darstellung gemäß FIG 5 zeigt das Ablaufebenenmodell der universellen Bewegungssteuerung. Die Priorisierung der Ebenen

wird wie im vorangegangenen durch einen Pfeil in Richtung zur höchsten Priorität angedeutet. Die niederpriorste Ebenengruppe ist die sogenannte "Ebenengruppe Hintergrund-Bearbeitung". Sie besteht aus einer "Systemebene Hintergrund", aus einer

5 "Anwenderebene freier Zyklus" und aus einer "Anwenderebene sequenziell". Die Tasks dieser drei gleichprioreren Ebenen (angedeutet durch die gestrichelten Grenzlinien) werden zyklisch im Round-Robin-Verfahren abgearbeitet. Eine auf die "Ebenen-

10 gruppe Hintergrund-Bearbeitung" höherprior folgende "Ablaufebene" ist eine vom Anwender anforderungsspezifisch frei projektierbare Anwenderebene FA, durch doppelte Umrandung gekennzeichnet, für Alarm- und/oder Event- und/oder Regelungs-

15 und/oder sonstige zyklische Tasks. Diese Anwenderebene FA besteht somit explizit aus vier Typen von Ebenen, die wiederum hinsichtlich ihrer Prioritäten innerhalb der Anwenderebene FA vom Anwender staffelbar sind.

Typ 1: Anwenderebene Event

20 Typ 2: Anwenderebene Alarm

Typ 3: Anwenderebene getaktet

Typ 4: Systemebene parametrisiert

25 Ebenen dieser Typen können vom Anwender frei wählbar innerhalb der Anwenderebene FA, mit jeweils zugrunde gelegten, vom Anwender vergebaren Prioritäten, angeordnet werden. Damit hat der Anwender die Möglichkeit, eine den Anforderungen und

30 Randbedingungen der Steuerungsaufgabe und des zu steuernden technischen Prozesses optimale Ausprägung der universellen Bewegungssteuerung zu erreichen.

In der "Anwenderebene Event" sind z.B. Tasks angeordnet, die

35 auf Eingänge der Peripherie reagieren. In der "Anwenderebene Alarm" sind z.B. Tasks angeordnet, die auf Grenzwertüberwachungen reagieren. In der "Anwenderebene getaktet" sind zyk-

lische anwenderprogrammierbare Tasks enthalten. In die "Systemebene parametriert" können von außen zuladbare Programme integriert werden. Dadurch ist es möglich, daß die universelle Bewegungssteuerung dynamisch um zusätzliche technologische Funktionalitäten erweitert werden kann. In diese "Systemebene parametriert" werden üblicherweise Tasks für langsame Reg-
5 lungs- bzw. Überwachungsaufgaben (z.B. Aufgaben mit Zykluszeiten im Bereich von 100 ms) zugeladen.

10 Die nächsthöherpriorie Ebene im Ablaufebenenmodell der universellen Bewegungssteuerung ist eine "Anwenderebene für asynchrone Fehler". In dieser Ebene kann der Anwender, ähnlich wie bei einer speicherprogrammierbaren Steuerung, die Behandlung von Fehlerzuständen ausprogrammieren. In der "Anwender-
15 ebene für asynchrone Fehler" sind z.B. Tasks angesiedelt, die auf technologische Alarime reagieren. Der Anwender hat auch die Möglichkeit, innerhalb dieser "Anwenderebene für asynchrone Fehler" eine für die Produktausprägung spezifische Anzahl von Ebenen zu parametrieren. Der Übersichtlichkeit hal-
20 ber sind Einzelheiten hierzu in der Darstellung nicht gezeigt. Der Anwender kann somit bedarfsweise bestimmten Fehlerereignissen eine bestimmte Priorität zuordnen.

25 Als nächstes folgt die "Event-Systemebene". Die Tasks der "Event-Systemebene" reagieren auf kritische interne oder externe Ereignisse, wie z.B. Nothalt.

Die nächste Ebene ist eine "Interpolatorebene". Sie enthält eine "getaktete Systemebene" und eine "Anwenderebene".

30 Die höchstpriorie Ebene ist die "Lagereglerebene". Auch sie enthält eine "getaktete Systemebene" und eine "Anwenderebene". Die Anwenderebenen der Lageregler- und Interpolatorebene enthalten Tasks, die im Lageregler- bzw. Interpolatortakt
35 aufgerufen werden. Die Laufzeit dieser Tasks wird überwacht, das Überschreiten einer durch das System festgelegten Zeit

führt zum Abbruch der Ebene und zum Auslösen eines asynchronen Fehlers in der "Anwenderebene für asynchrone Fehler".

5 Der Lageregler hat eine höhere Priorität als der Interpolator, d.h. der Lageregler kann nicht vom Interpolator unterbrochen werden, wobei der Lageregler aber den Interpolator unterbrechen kann.

10 Im Ablaufebenenmodell der universellen Bewegungssteuerung können prinzipiell innerhalb der einzelnen Ablaufebenen neben den bereits erwähnten, weitere priorisierende Schichtungen vorgesehen sein.

15 Die Darstellung gemäß FIG 6 zeigt als OO-Strukturdiagramm, wobei die Kardinalitäten durch eine gängige Ziffern-Notation angezeigt werden, ein Technologiepaket TP mit seinen Bestandteilen:

20 a) Ablauffähige Code-Teile (Code)

b) Parameter (PAR)

c) Firmware-Konfiguration (FWK)

25 d) Mindestens einem Technologieobjektyp (TO)

e) Sprachmechanismen (SPR)

30 f) Deklarations- und Beschreibungsteil (ACC)

35 Die 1 bis n Code-Teile (z.B. C-Funktionen) werden beispielsweise für die Bewegungsführung oder die Lageregelung oder für eine andere Technologie verwendet. Die Code-Teile können unter anderem Befehle für Temperaturführung, Temperaturregelung oder für spezielle Technologien wie z.B. Pressen oder Kunststoffverarbeitung beinhalten. Wie diese Code-Teile im Ablaufebenenmodell der Steuerung in die Systemebenen eingehängt

werden und in welcher Reihenfolge sie zur Abarbeitung, d.h. Ausführung, gelangen sollen, wird in der Firmware-Konfiguration FWK festgelegt. In dieser steht also die Information, in welche Systemebene ein Code-Teil integriert werden soll und
5 wenn in einer Systemebene mehrere Code-Teile integriert sind, in welcher Reihenfolge diese Code-Teile abgearbeitet werden sollen.

Der Parameter-Teil PAR beinhaltet Oberflächen- und Bedienparameter für das Engineering-System (ES; FIG 1, FIG 4, FIG 8)
10 als auch die Mechanismen für das Run-Time-System (RTS; FIG 1, FIG 4, FIG 8), die eine Parametrierung ermöglichen. Damit hat der Anwender die Möglichkeit, Instanzen von Technologieobjekttypen TO eines Technologiepaketes TP gemäß seinen Anforderungen zu parametrieren.
15

Mit Hilfe der 1 bis n Sprachmechanismen SPR eines Technologiepakets TP läßt sich der Sprachvorrat des Engineering-Systems (ES; FIG 1, FIG 4, FIG 8) um Befehle und Operatoren,
20 die für das zugrundeliegende Technologiepaket TP mit seinen zugehörenden 1 bis n Technologieobjekttypen TO adäquat und sinnvoll sind, erweitern. Sprachmechanismen SPR müssen ins Engineering-System (ES; FIG 1, FIG 4, FIG 8) und ins Run-Time-System (RTS; FIG 1, FIG 4, FIG 8) der Steuerung geladen werden. Nachdem solche Sprachmechanismen (z.B. "erhöhe Temperatur") im Engineering-System (ES; FIG 1, FIG 4, FIG 8) installiert wurden, sind sie im Compiler und in der Oberfläche bzw. im Browser des Engineering-Systems (ES; FIG 1, FIG 4, FIG 8) bekannt und können vom Anwender in seinen Anwenderprogrammen direkt verwendet werden. Durch eine Plug & Play-Technologie wird sichergestellt, daß im Engineering-System (ES; FIG 1, FIG 4, FIG 8) bekannte Sprachmechanismen auch als ablauffähiges Codestück im Run-Time-System (RTS; FIG 1, FIG 4, FIG 8) vorhanden sind. Der Anwender verwendet also die
30 Spezifikation der Sprachmechanismen, um die Implementierung im Run-Time-System (RTS; FIG 1, FIG 4, FIG 8) braucht er sich nicht mehr zu kümmern. In der später noch behandelten FIG 8
35

wird das Zusammenspiel von Zuladen, Verwenden und Abarbeiten von Sprachmechanismen SPR von Technologiepaketen TP genauer erläutert werden.

- 5 Zurück zu FIG 6: In der ACC-Komponente eines Technologiepakets TP befindet sich die Beschreibung aller Sprachelemente, die das Technologiepaket TP enthält, die Beschreibung aller Systemvariablen und aller Typen, die im Technologiepaket TP verwendet werden. Die ACC-Komponente entspricht somit einem
- 10 Deklarations- und Beschreibungsteil für das Technologiepaket TP. Diese ACC-Komponente wird in erster Linie ins Run-Time-System (RTS; FIG 1, FIG 4, FIG 8) der Steuerung geladen. Dadurch wird sichergestellt, daß sich alle Informationen bzgl. vorhandener Technologiepaketen TP und Technologieobjekttypen
- 15 TO im Run-Time-System der Steuerung befinden und somit der Anschluß von Bedien- und Beobachtungsgeräten (z.B. Operator Panels) sehr leicht möglich ist.

- 20 Nachfolgende Tabelle zeigt, wohin die Bestandteile des Technologiepakets TP innerhalb der Steuerung geladen werden: entweder ins Engineering-System (ES; FIG 1, FIG 4, FIG 8) oder ins Run-Time-System (RTS; FIG 1, FIG 4, FIG 8) oder sowohl ins Engineering-System (ES; FIG 1, FIG 4, FIG 8) als auch ins Run-Time-System (RTS; FIG 1, FIG 4, FIG 8).

25

Bestandteil	TP-Komponente	wird geladen in:
ablauffähiger Code	Code	RT
Parametrieroberfläche (und entsprechendes Wissen über die einzelnen Parameter)	PAR	ES
Konfigurations-Informationen (Informationen, wie und wo die Teile in das Ablaufsystem eingeklinkt werden)	FWK	ES + RT
Anwenderschnittstelle (Befehle (MOVE, POS, ...), SFCs, SFBs, Systemvariablen, ...)	SPR	ES + RT
Anwenderschnittstelle (graphische Informationen)	SPR	ES
Beschreibungsinformationen für Systemvariablen, Alarmer, ...	ACC	ES + RT
Objekttypen (Technologische Objekte)	TO	ES + RT
Versionsinformationen für Konsistenz zwischen RT, Paketen und Objekten	ACC	ES + RT

In der Darstellung gemäß FIG 7 werden als OO-Strukturdiagramm exemplarisch mögliche Technologieobjekttypen (TO; FIG 6) für

ein Technologiepaket (TP; FIG 6) Kunststoff TPK dargestellt. Bei der Kunststoffbearbeitung bzw. -erzeugung benötigt man üblicherweise eine Temperaturregelung und eine Druckregelung. Der Druck, der dann auch durch den Druckregler DR geregelt werden muß, wird üblicherweise durch eine einfache Achse A aufgebaut, indem die Achse die Materialpaste zusammenpreßt. Für die Temperaturregelung sind in diesem Beispiel zwei Temperaturregler vorgesehen, ein schneller Temperaturregler TRS und ein langsamer Temperaturregler TRL. Wie aus dem OO-Strukturdiagramm ersichtlich, leiten sich der langsame Temperaturregler TRL und der schnelle Temperaturregler TRS aus dem allgemeinen Temperaturregler TR ab. Die beiden Temperaturregler TRS und TRL, der Druckregler DR und die Achse A werden in dem vorliegenden Technologiepaket Kunststoff durch vier Technologieobjekttypen TO repräsentiert, nämlich TRS, TRL, DR und A. Durch die Kardinalität (Ziffer 1) wird angedeutet, daß bei diesem Beispiel genau ein schneller TRS und ein langsamer Temperaturregler TRL, sowie genau ein Druckregler DR und eine Achse A verwendet werden. Hinter dem schnellen Temperaturregler TRS kann sich z.B. ein PID-Regler, hinter dem langsamen Temperaturregler TRL kann sich z.B. ein P-Regler verbergen, das sind aber Implementierungsdetails, die einen Anwender der Funktionalitäten dieser Technologieobjekttypen im Engineering-System (ES; FIG 1, FIG 4, FIG 8) verwendet, nicht zu interessieren brauchen. Ein Anwender kann somit die Funktionalitäten dieser Technologieobjekttypen (TO; FIG 6) im Engineering-System (ES; FIG 1, FIG 4, FIG 8) verwenden, ohne sich um Implementierungsdetails kümmern zu müssen.

In der Darstellung gemäß FIG 8 wird gezeigt, daß aus einer Datenquelle D, die Beschreibungsinformationen für Systemvariablen sowie gegebenenfalls Alarmer und/oder Sprachbefehle enthält, über den Informationspfad I4 an den Umsetzer U übermittelt werden. Der Umsetzer U generiert aus den eingespeisten Daten Parametrierinformationen für das Engineering-System ES4, für das Ausgabemedium AM und für das Run-Time-System RTS4. Der Umsetzer U wird vor der Compilierung der Run-Time-

bzw. Engineering-System-Software aufgerufen. Er erzeugt Sourcen, die dann bei der Erzeugung der Run-Time- bzw. Engineering-System-Software compiliert werden. Auf dem Informationspfad I5 werden vom Umsetzer U die Parametrierinformationen zum Engineering-System ES4 transferiert. Die vom Umsetzer U erzeugten Parametrierinformationen für die Dokumentation werden über den Informationspfad I6 an das Ausgabemedium AM, in FIG 8 dargestellt durch einen Drucker, weitergeleitet. Als Ausgabemedium ist aber auch beispielsweise eine Online-Hilfe am Bildschirm vorstellbar. Die vom Umsetzer generierte Parametrierinformation für das Run-Time-System wird über den Informationspfad I7 in das Ablaufebenenmodell AE des Run-Time-Systems RTS4 geladen.

Der Umsetzer U wird immer dann aufgerufen, wenn er aus der Datenquelle D über den Informationspfad I4 neue Parametrierinformation erhält. Der Umsetzer U erzeugt in einem solchen Fall neue Sourcen für die Dokumentation, das Run-Time-System RTS4 und für das Engineering-System ES4.

Beim Betrieb der Steuerung werden die ins Engineering-System ES4 geladenen Systemvariablen vom Run-Time-System RTS4 über den Informationspfad I8 mit aktuellen Daten des technischen Prozesses versorgt. Der Anwender hat die Möglichkeit, beziehungsweise auf den aktuellen Zustand des technischen Prozesses (TP1, TP2; FIG 1 bzw. FIG 4) weitere Eingaben am Engineering-System ES4 durchzuführen.

Über den Informationspfad I9 kann das Run-Time-System RTS4 ein Gerät zum maschinennahen Überwachen und Steuern (in FIG 8 dargestellt durch ein Operator Panel OP) mit Informationen (z.B. Alarmen) versorgen.

Der Umsetzer U tritt beim Betrieb der Steuerung nicht mehr in Erscheinung. Die Informationspfade I4, I5, I6 und I7 werden bei der Erzeugung der Steuerungssoftware verwendet, aber

nicht beim Betrieb der Steuerung. Beim Betrieb der Steuerung werden die Informationspfade I8 und I9 verwendet.

Patentansprüche

1. Universelle Bewegungssteuerung mit Engineering- und Run-Time-System, welche funktionell die klassischen Aufgaben einer speicherprogrammierbaren Steuerung und einer numerischen Steuerung in sich vereinigt,

5 d a d u r c h g e k e n n z e i c h n e t,
daß ein einheitliches Ablaufebenenmodell (AE) dergestalt gebildet ist, daß es mehrere Ablaufebenen unterschiedlichen
10 Typs mit unterschiedlicher Priorität aufweist, wobei von
höchster bis niedrigster Priorität verschiedene Anwender- und
Systemebenen vorgesehen sind und daß jeweils Technologiepakete (TP) anwenderseitig in das Engineering- und/oder Run-Time-System (ES1-ES4, RTS1-RTS4) ladbar sind, daß eine Datenquelle
15 (D) für Beschreibungsinformationen für Systemvariablen sowie
gegebenenfalls Alarmer und/oder Sprachbefehle über einen Umsetzer (U) dem Engineering-System (ES1-ES4) Sprachbefehle
und/oder Systemvariablen zur Verfügung stellt, daß aus dem
Run-Time-System (RTS1-RTS4) die Systemvariablen mit aktuellen
20 Daten des technischen Prozesses (TP1, TP2) versorgbar sind
und daß über eine Bedienoberfläche des Engineering-Systems
(ES1-ES4) weitere Eingaben anwenderseitig machbar sind.

2. Universelle Bewegungssteuerung nach Anspruch 1,

25 d a d u r c h g e k e n n z e i c h n e t,
daß vom Umsetzer (U) aus dem Bestand der Datenquelle (D) relevante Dokumentationsinformationen an ein Ausgabemedium (AM) weiterleitbar sind.

30 3. Universelle Bewegungssteuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,
daß als Ablaufebenen vorgesehen sind:

- a) eine Lagereglerebene, bestehend aus zugehöriger getakteter Systemebene und Anwenderebene,
 - b) eine Interpolatorebene, bestehend aus zugehöriger getakter Systemebene und Anwenderebene,
 - 5 c) eine Event-Systemebene für reaktionspflichtige Ereignisse,
 - d) eine Anwenderebene für asynchrone Fehler
 - e) eine weitere, vom Anwender anforderungsspezifisch frei projektierbare Anwenderebene (FA) für Alarm- und/oder E-vent- und/oder Regelungs- und/oder sonstige zyklische
 - 10 Tasks,
 - f) eine aus der Abfolge von Bewegungssequenzen, freien Zyklen und niederprioren sonstigen Systemtasks, gebildete Ebenengruppe für Hintergrund-Bearbeitung,
- wobei die Ablaufebenen a bis e eine Ebenengruppe für Echt-
- 15 zeit-Bearbeitung bilden.

4. Universelle Bewegungssteuerung nach einem der vorstehenden Ansprüche,

d a d u r c h g e k e n n z e i c h n e t,

20 daß die Technologiepakete (TP) beinhalten:

- a) Code-Teile, die die Regelungsspezifika für das Run-time-System (RTS1-RTS4) repräsentieren und
- b) einen Konfigurierteil (FWK) der die Zuordnung dieser Code-Teile zu den jeweiligen Systemebenen, sowie deren Bearbeitungsfolge aufweist, wobei
- 25 c) bedarfsweise diese Informationen des Konfigurierteils (FWK) auch an das Engineering-System (ES1-ES4) weiterleitbar sind.

30 5. Universelle Bewegungssteuerung nach Anspruch 4,

d a d u r c h g e k e n n z e i c h n e t,

daß die Informationen des Konfigurierteils (FWK) eines Technologiepaketes (TP) über die Datenquelle D und den Umsetzer U

ins Run-Time-System (RTS1-RTS4) und ins Engineering-System (ES1-ES4) gebracht werden.

5 6. Universelle Bewegungssteuerung nach Anspruch 4 oder 5,
d a d u r c h g e k e n n z e i c h n e t,
daß jedes Technologiepaket (TP) eine angepaßte Anzahl von
Technologieobjekttypen (TO) für das Run-Time-System (RTS1-
RTS4) beinhaltet.

10 7. Universelle Bewegungssteuerung nach Anspruch 4, 5 oder 6,
d a d u r c h g e k e n n z e i c h n e t,
daß des weiteren Bedienoberflächeninformationen, insbesondere
Bedienparameter (PAR), und/oder Sprachmechanismen (SPR)
und/oder Deklarationsteile (ACC) den Code-Teilen zuweisbar
15 sind.

Zusammenfassung

Universelle Bewegungssteuerung

- 5 Die in einer universellen Bewegungssteuerung UMC benötigten Parametrierinformationen (Beschreibungen von Systemvariablen, Alarmen und Sprachbefehlen) werden über einen zentralen Um-
- 10 setzer (U) aus einer einheitlichen Beschreibungssprache generiert und an Engineering-System (ES1-ES4), Run-Time-System (RTS1-RTS4) sowie Ausgabemedien (AM) für die Dokumentation verteilt. Dadurch ist gewährleistet, daß die Parametrierdaten für alle Systemteile konsistent sind. Darüberhinaus können
- 15 auch Konfigurierinformationen (FWK) von Technologiepaketen (TP) über den Umsetzer U ins Engineering- (ES1-ES4) und ins Run-Time-System (RTS1-RTS4) der Steuerung gebracht werden.

FIG 8

1/5

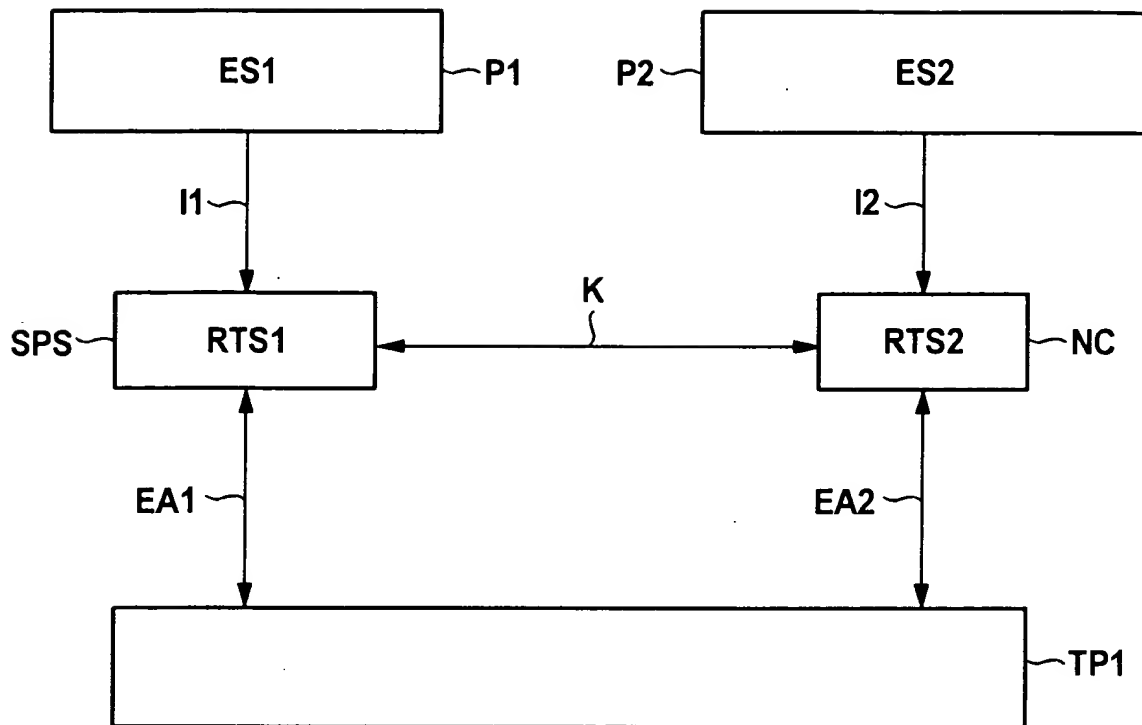


FIG 1

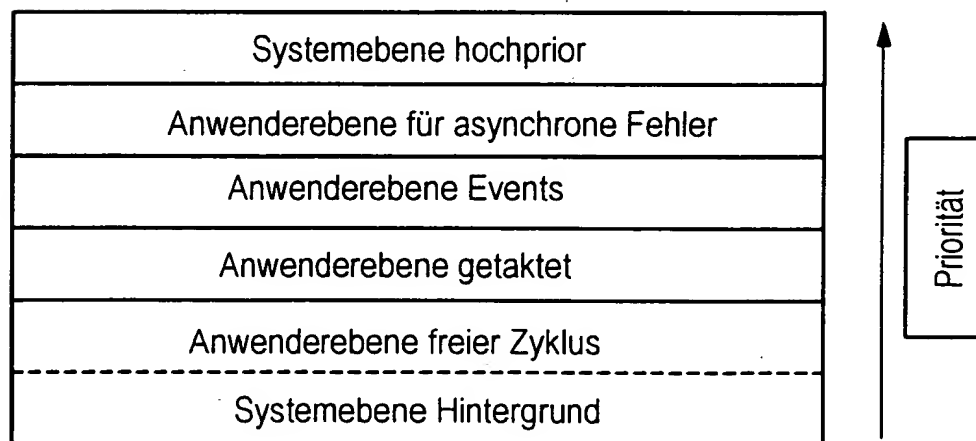


FIG 2

2/5

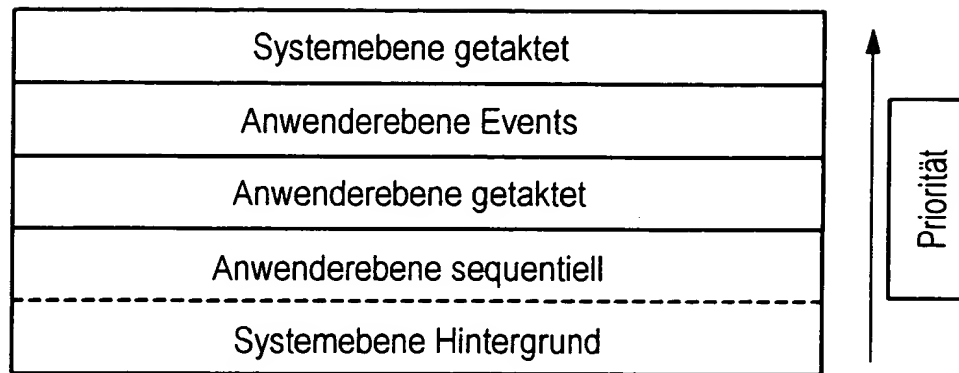


FIG 3

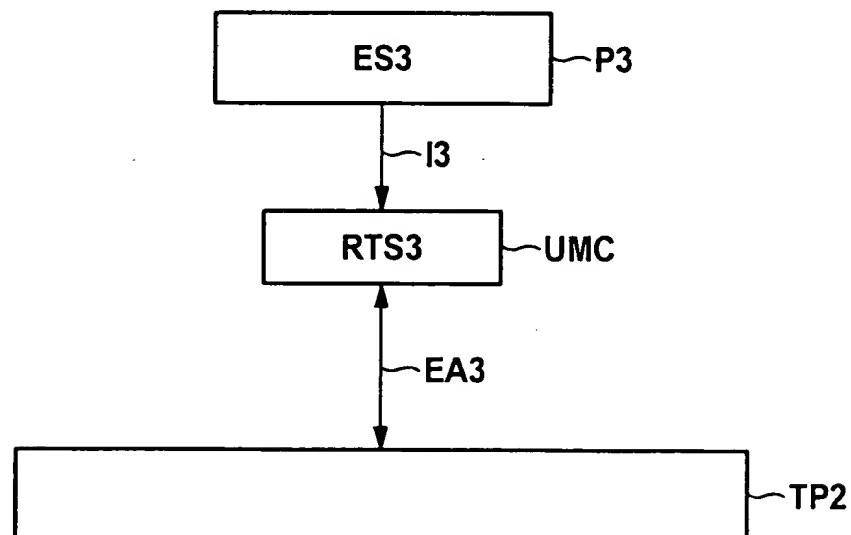


FIG 4

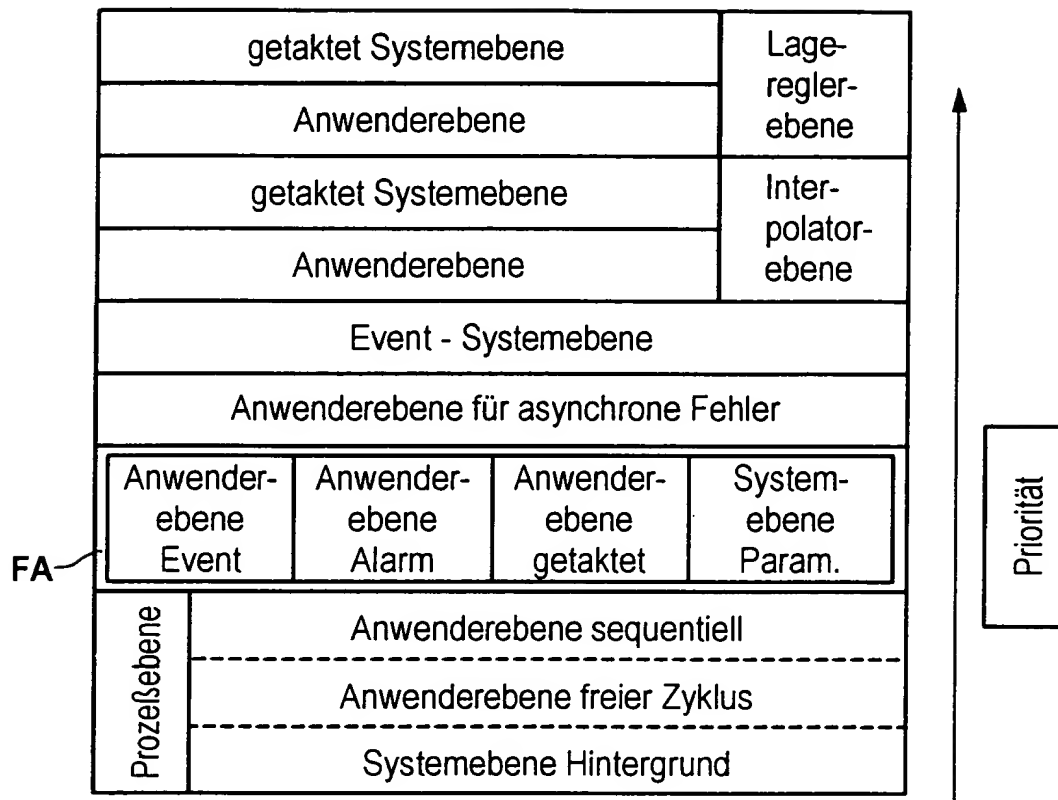


FIG 5

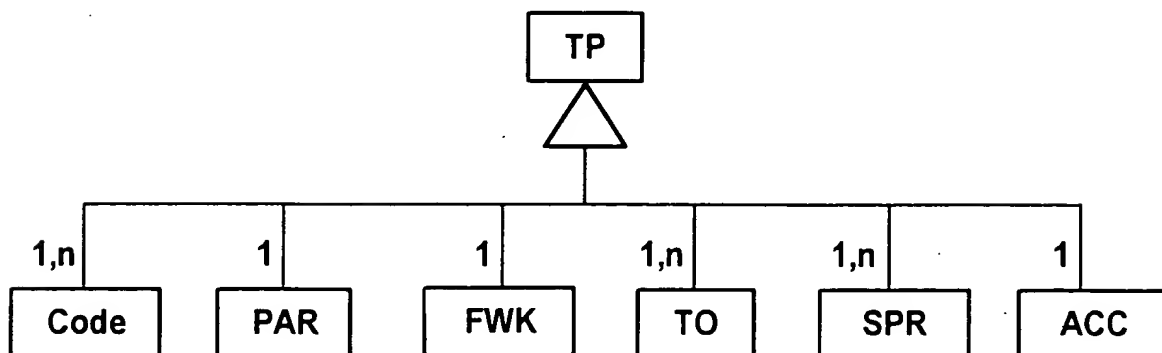


FIG 6

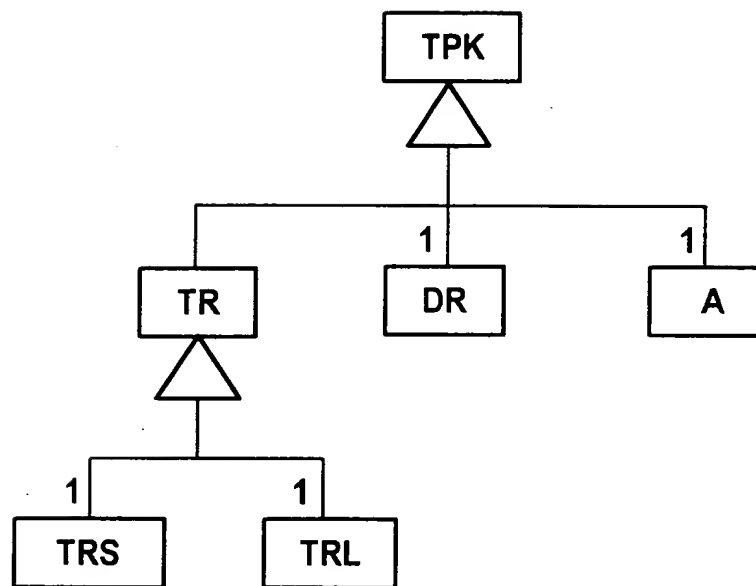


FIG 7

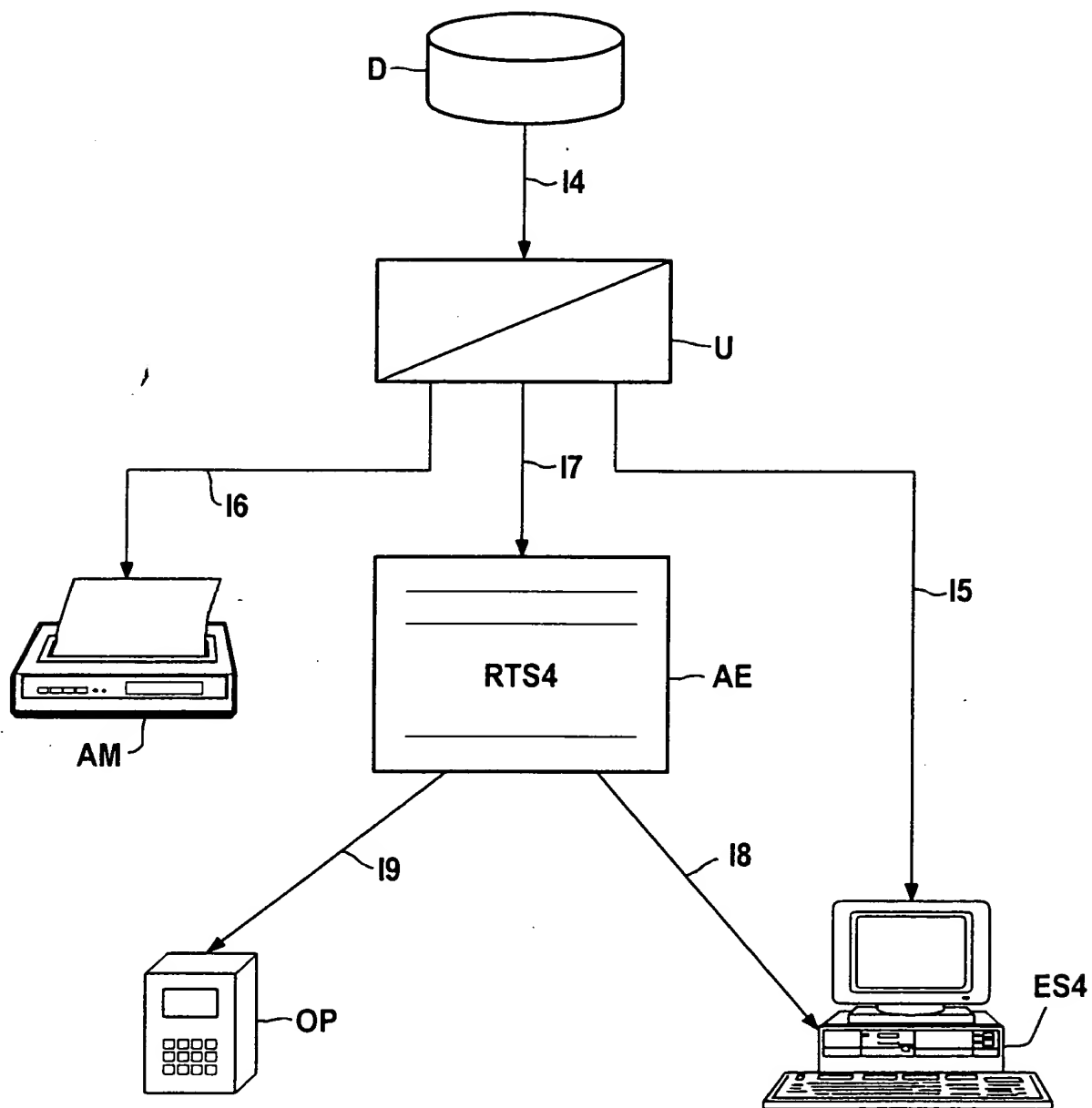


FIG 8